Amendments to the Specification:

Please replace paragraph [0002] with the following amended paragraph:

[0002]This invention relates to generation, from graphically produced description of workflow process, of source code and ~~building~~ compiling and linking instructions of fully executable transactional workflow application with adaptive high-performance capacity for transactional ~~workload~~ processing of concurrent workflow and its real-time visualization.

Please replace paragraph [0006] with the following amended paragraph:

[0006]Customer-facing e-business applications are expected to ~~satisfy two requirements. First one is to~~ respond in real-time to customer requests, ~~request. Second one is to have a capacity~~ to initiate and conduct simultaneous processing of unpredictably high number of user interactions, and be process-centric, i.e. capable of executing set of related actions structured as transactional workflow. ~~asynchronously arriving requests. Buffering asynchronously arriving requests in queues and subsequent synchronous processing solves the problem with requests spikes. Buffered requests, however, will not be responded in real-time. High responsiveness of customer-facing e-business systems, e.g. short time interval between sending a request and receiving a response with result of request execution, is already obligatory requirement. In parallel with increase in future volumes of e-business transactions, difficulties associated with providing high responsiveness of e-business systems is expected to sharpen.~~ Eventual execution as single transaction, instead as transactional workflow, of all actions related to responding to a step of single user interaction will hold database locks until the transaction is completed or aborted. As a result, even when multiple instances of the same e-business application run concurrently, these instances will perform sequentially - one will hold database locks and all other will wait for database locks to be released. Therefore, the only way to cope with both increased number of concurrent users and increased processing complexity of user interactions while preserving real-time responsiveness is to use workflow systems with capacity for processing of high-volume concurrent transactional workflow.

Please replace paragraph [0007] with the following amended paragraph:

[0007]Ability to adapt to changes in environment, competition, and customer needs ~~means ability of corporation to transform itself rapidly. Such transformation~~ implies high-speed of development of e-business systems ~~development process~~ and flexibility of existing e-business systems for future fine-grain changes within limited time. The most ~~generic and~~ flexible solutions, created with known technologies, satisfying those two requirements, are in form of workflow engine, workflow script and workflow activities' software components ~~being able to concurrently run multiple workflow applications~~. A workflow script ~~application that will run on an engine~~ is a form of description of workflow graph representing sequence of execution of workflow activities ~~and software component representing those activities~~. During workflow application running, workflow engine interprets workflow graph description and this results in execution of software components, representing workflow activities, in described sequence. If modifications of a workflow process are needed, desired results are achievable with changes in workflow graph, or components, or both. United States Patent Documents 20020040339, 20020065701, 20020170035, and 20020184616 demonstrate prior art related to concept of workflow engines.

Please replace paragraph [0008] with the following amended paragraph:

[0008]Such flexibility, however, has its price. The price is low efficiency of consumed resources, low processing speed, and lack of concurrency. Even if a hypothetical generic solution, in form of ~~an~~ workflow engine, script and components ~~being able to run one or more concurrent workflow processes with different workflow configurations~~, has ~~the~~ ability for concurrent processing of multiple requests, no such solution can compete in respect to ~~simplicity~~ efficiency and speed with a proprietary executable application, designed and built for a particular workflow process. ~~One of consequences that should be expected with an increase in simplicity of workflow application is an increase in its efficiency and speed. Another factor that will~~ What may contribute to increasing workflow application efficiency and speed ~~will be~~ is any partial or full replacement of interpreted code with executable ~~code loadable in computer memory code segments for invoking and execution~~. Replacement of interpreted code with executable is the approach used with some Java Virtual Machine (JVM) based application servers. The performance improvement, however, does not extend over workflow concurrency because of the inherent

concurrency limitations of JVM. With JVM application servers, the only way to process 2 workflow requests concurrently is to concurrently run 2 workflow application instances.

Please replace paragraph [0017] with the following amended paragraph:

[0017]Preferred embodiment of invention's methods ~~and article of manufacture for~~ related to graphical development of transactional workflow application source code and building instructions is in form of an EXE file and two DLLs. This is invention's first embodiment. One of its DLLs contains components for interactive graphical design of workflow processes and for ~~production of a description~~ producing definition of designed workflow process from graphical objects representing the designed workflow process. The other DLL contains components producing custom source code and compiling and linking instructions, sufficient for building transactional workflow applications in form of fully executable code, from a description of a workflow process.

Please replace paragraph [0018] with the following amended paragraph:

[0018]Preferred embodiment of invention's ~~methods and article of manufacture for~~ system with adaptive high-performance capacity for processing of concurrent transactional workload ~~processing~~ is in form of three DLLs. This is invention's second embodiment. Two of its DLLs contain components with functionality enacting the high-performance capacity ~~in~~ for concurrent transactional workload processing of applications created with invention's first embodiment. Third DLL contains components for real-time visualization of workload processing.

Please replace paragraph [0019] with the following amended paragraph:

[0019]This invention overcomes the limitations of known prior art methods for development of custom made application source code for fully executable workflow applications by providing a formal method of development of software source code for fully executable workflow applications. ~~This formal method is used to build the article of manufacture for generation of source code and compiling and linking instructions, sufficient for building transactional workflow applications in form of fully executable code.~~ In the preferred embodiment, the generated source code is in

attributed Visual C++ and the generated compiling and linking instructions are in form of Microsoft Visual Studio .Net solution (.sln) and project (.proj) files.

Please replace paragraph [0020] with the following amended paragraph:

[0020] Transactional workflow applications ~~in form of executable code~~, created with invention's first embodiment and using at run-time invention's second embodiment, are different from transactional workflow applications created with prior art ~~systems, that are any form of scripts run on workflow engines~~ in regard to ability for high-volume concurrent processing. ~~The difference is in applications' ability to provide performance capacity in transactional workload processing higher then performance capacity of applications in form of scripts run on workflow engines.~~ Transactional workflow applications created with prior art have inherent concurrency limitations. As Web-accessible business applications must have capacity to serve unpredictably high number of concurrent users, concurrency limitation is a barrier in creation of Web applications processing user interactions with execution of transactional workflow such as, for example, Web-accessible facility for online negotiation of electronic purchases. This invention addresses the need for high-volume concurrent workflow by enabling concurrent processing of 1,000 workflow instances per computer CPU.

Please replace paragraph [0021] with the following amended paragraph:

[0021] This invention's method ~~and article of manufacture~~ for graphical development of source code for fully executable workflow applications is different from prior art non-standardized methods for development of software for executable workflow applications. The difference is best demonstrated by ~~in~~ the speed of software development process. The speed of software development with the invention's first embodiment is as fast as the speed of development of workflow applications in form of workflow scripts. No workflow application in form of script, however, can possess a capacity for more then just a very limited concurrency.

Please replace paragraph [0022] with the following amended paragraph:

[0022] This invention's ~~method and article of manufacture for constructing a configuration~~ system for transactional processing of workflow with a hierarchy of class objects ~~and threads with capacity for high-performance concurrent processing~~

~~of multitude of transactional workflow requests of identical type~~ is different from known prior art system for transactional processing of workflow ~~articles of manufacture with capacity for high-performance concurrent processing of multitude of transactional workflow requests of identical type~~. The main difference is in flexibility of ~~configuration~~ hierarchy of class objects ~~and threads~~ to represent a non-limited number of workflow graphs and thereby to enact a non-limited number of workflow processes. ~~The second difference, resulting from this flexibility, is the speed of development of transactional workflow applications with capacity for high-performance. The development time is only a small fraction of development time needed with prior art articles of manufacture.~~

Please replace paragraph [0024] with the following amended paragraph:

[0024]This invention overcomes limitations of known prior art systems for transactional processing of workflow in their ability to counteract development of software related bottlenecks ~~in workflow applications due to~~ as result of variations of working conditions ~~of~~ in distributed application environment. Neutralization of development of software related bottlenecks is enacted by apparatus ~~providing a method and article of manufacture~~ ensuring that: workflow system ~~application~~ is supplied with sufficient number of threads for ~~high-performance in~~ processing of transactional workload when an insufficiency is generated by ~~variable working conditions of~~ communication delays in distributed application environment; and no thread is overloaded with dispatching or supervising work.

Please replace paragraph [0025] with the following amended paragraph:

[0025]This invention overcomes limitations of known prior art systems, in their ability to visualize workflow application's thread structures, threads quantity, and threads usage~~, by providing a method and article of manufacture for real-time visualization of application's thread structures, threads quantity, threads usage, and scaling-enacted changes in threads structure and quantity~~. Facilitation of real-time visualization will empower workflow application administrators with abilities to evaluate application workload, to detect points of delay caused by distributed infrastructure, and to observe application's scaling-related actions.

Please replace paragraph [0058] with the following amended paragraph:

[0058] This invention is about engineering approach ~~to software development~~ and ~~speed:~~ speed of development and future modifications of workflow applications~~, speed of future application modification, and most importantly – resulting products' ability to~~ and about enabling execution of ~~perform~~ large number of concurrent business interactions ~~of similar type with high~~ ~~speed~~.

Please replace paragraph [0059] with the following amended paragraph:

[0059] Invention provides method ~~and article of manufacture~~ for graphical development of fully executable workflow applications. Application's graphical development has two phases: application design phase, where definition of a workflow process is interactively produced, and build phase, producing executable code of a workflow application that will perform in a way described by the definition of workflow process.

Please replace paragraph [0079] with the following amended paragraph:

[0079]Fig. 8 is block diagram of graphical development of a workflow application ~~according to article of manufacturing being described~~. Using a tool for graphical development of workflow applications ~~based on this article of manufacture~~, a workflow application architect interactively executes workflow designer 801 and produces a definition of a workflow process 802. Execution of software generators 803 takes as input a data set being a definition of a workflow process 802. Result of execution of software generators 803 includes: source code 804 of workflow application that after being compiled, linked, installed, and executed will behave according to the definition of a workflow process 802; compiling instruction 805 necessary for compiling the generated source code 804; linking instructions 806 for producing application's executable code from results of execution of compiling instructions 805. Software source code generation is result of both analysis of description of desired workflow and knowledge of the ~~article of manufacturing being~~ embodiment of the rest of this invention. Generated software source code is responsible for customizable part of workflow application, dealing with application initialization and termination. The customizable part of application initialization facilitates construction of hierarchical trees of objects and hierarchical structure of threads. Source code generation involves usage of source code templates, class

templates, and pure generation of source code by created for this purpose program executable code, where elements of workflow definition are passed as input parameters of execution of this executable-code-provided functionality. Once results of execution of software generators 803 are obtained, compilers 807 are executed according to compiling instruction 805 taking as compilers' input generated source code 804 and results of compiling 808 are provided as input to linker execution 809 according to linking instructions 806 to produce workflow application executable code 810.

Please replace paragraph [0081] with the following amended paragraph:

[0081]Invention further includes a system ~~provides a method and article of manufacture producing configuration of class objects and threads~~ with capacity for concurrent processing of multitude of transactional workflow requests of identical type. Core of this system ~~method and article of manufacture~~ is ~~construction of~~ hierarchy ~~tree~~ of class objects with capacity to represent variety of workflow configurations and associated structure of threads with capacity for concurrent processing of multitude of workflow requests. In preferred embodiment, the workflow control-connectors and the not-mentioned-yet workflow notification-connectors are implemented as transactional MSMQ messages.

Please replace paragraph [0098] with the following amended paragraph:

[0098]Invention's ~~method and article of manufacture with~~ capacity for concurrent processing of ~~multitude of transactional~~ workflow ~~requests of identical type relates to a~~ is enabled by hierarchical structure of threads with four levels ~~and thread levels relations and interaction~~. The hierarchical structure levels below top level are organized as multitude of horizontally arranged divisions. Each division is autonomous and self-contained in conducting its tasks. Top level thread is responsible for making adaptive decisions, and executing and supervising adaptive behavior related to allocation and de-allocation of system resources based on its own assessment of application needs and goals. The provided by threads of the hierarchical structure capacity for concurrent processing of multitude of requests is limited only by environmental factors such as availability of reserve of system memory and unused CPU power and ability of networking infrastructure to cope with generated traffic.

Please replace paragraph [00107] with the following amended paragraph:

[0107]Invention further includes a method ~~and article of manufacture~~ for transactional plugging of software components into workflow-process. The preferred embodiment of this method ~~article of manufacture~~ uses services of COM+. It involves a Non-Transactional component and a Transactional component installed on a physical computer as a COM+ application and software components that will be ~~transitionally~~ plugged into workflow process, installed on the same physical computer as a COM+ application. COM+ provides component-intercepting objects and functionality of just-in-time-activation for the Transactional component and for software components that will be ~~transitionally~~ plugged into workflow process. The just-in-time-activation functionality ensures that when a method of a component-intercepting object is called, it will instantiate transactional object, call its method, vote for transaction on method exit, and destruct the transactional object. The Non-transactional component is a special kind of data holding component. The data it holds is required for execution of all future request-processing and request-forwarding transactions that will be performed by the processing thread. This data however is not specific or distinctive to any particular request. The main reason for existence of the non-transactional component is to prevent the processing thread from multiple transmissions of the mentioned, non-specific to any particular request, data over COM+ application process boundaries and multiple creations of the intercepting objects.

Please replace paragraph [00111] with the following amended paragraph:

[0111]The capacity for concurrent processing of multitude of ~~transactional~~ workflow requests of identical type is achieved by the hierarchical structure of threads with four levels and by ~~set of methods and article of manufacture specifying interactions and~~ division of labor between threads of hierarchical structure levels. ~~The first in that set is the method and article of manufacture for workload balancing structured at two levels.~~ Upper level of workload balancing comprises multitude of associations between a dispatching thread and multitude of supervising threads and involves dispatching thread, balancing workload between its associated supervising threads. Lower level of said workload balancing comprises multiple groupings of processing threads in pools associated with a supervising thread per pool and involves supervising threads balancing workload between processing threads of their associated pools. The rationale behind workload balancing with two levels is the need

to distribute the work, related to assignment of workload to processing threads and supervision of workload execution, between more threads and thereby to prevent threads responsible for this work from becoming workflow bottlenecks.

Please replace paragraph [00114] with the following amended paragraph:

[0114]This invention provides ~~a method~~ apparatus for software bottlenecks' prevention. Software bottlenecks' prevention involves encapsulation of a thread pool containing fixed number of processing threads with a supervising thread in a processing-pipe. Number of processing threads per processing-pipe is experimentally selected in a way ensuring that, without modifying threads' priorities, even if processing threads conduct full-scale dummy workflow work, processing-pipe's supervising thread will still have enough capacity to cope with its duties. Having number of processing thread fixed guarantees that supervising thread will never be overloaded with work leading to existence of workload dispatched to it and waiting to be processed and at the same time existence of number of IDLE processing threads in its pool.

Please replace paragraph [0116] with the following amended paragraph:

[0116] Software bottlenecks' neutralizing comprises construction of additional processing-pipes and inclusion of constructed additional processing-pipes in workload balancing process related to workflow-activity where development of bottleneck has been detected. Regulation of available workflow processing capacity includes ~~a method and~~ apparatus for automatic detection of conditions requiring workflow application scaling up and for automatic execution of workflow application scaling up. The automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of idle threads in processing-pipe's pool reached its critical minimum. Application scaling up is automatically triggered at a particular workflow-activity to counteract development of a bottleneck at that particular workflow-activity and automatically triggered at all application's workflow-activities for higher application responsiveness when workload increases. Application scaling up might require creation of an additional processing-pipe. In this case it cannot be automatically triggered. If there is an available non-activated processing pipe, automatically triggered scaling up involves activation of an additional

processing-pipe and inclusion of said additional processing-pipe in workload balancing scheme.

Please replace paragraph [0122] with the following amended paragraph:

[0122]Regulation of available workflow processing capacity further includes ~~method and~~ apparatus for automatic detection of conditions requiring workflow application scaling down and for automatic execution of workflow application scaling down. Automatic detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of busy threads in processing-pipe's pool reached its critical minimum. Application scaling down is automatically triggered to counteract a detected inefficiency in use of system memory and CPU time slice allocated to application threads.

Please replace paragraph [0126] with the following amended paragraph:

[0126]Final element of this invention is its system ~~method and article of manufacture~~ for real-time visualization of quantity, structure, and utilization of hierarchical structure of threads as they participate in processing of workflow requests. Threads forming the upper two levels of application's hierarchical structure of threads are not interesting for visualization since their quantity, type and status does not change after application is activated and before application is terminated. There is always one application supervising thread and total number of dispatching and synchronizing-dispatching threads is equal to number of workflow-activities. Additionally, where application's main flow of control shows more then one control-connectors having common destination node, the workflow-activity represented by this node will have synchronizing-dispatching thread, otherwise it will have dispatching thread. Threads forming First and Second levels of hierarchical structure of threads is where run-time changes happen and therefore hierarchical structure's adaptation-behavior-enacted modifications of First and Second levels are important for visualization. Visualization might be used as indicator of workload, indicator of points of delay caused by distributed infrastructure, and for observation and analysis of adaptive behavior of hierarchical structure of threads.